```
========= ORIGINAL JAVA CODE =========

import java.io.IOException;
import java.net.InetSocketAddress;
import com.sun.net.httpserver.HttpServer;
import com.sun.net.httpserver.HttpContext;
import com.sun.net.httpserver.HttpHandler;
import com.sun.net.httpserver.HttpExchange;
import java.io.OutputStream;
import java.nio.file.Files;
import java.io.File;

public class WebServer {
    public static void main(String[] args) throws IOException {
        HttpServer server = HttpServer.create(new InetSocketAddress(8000), 0);
        HttpContext context = server.createContext("/");
        context.setHandler( new HttpHandler() {
                @Override
                public void handle(HttpExchange exchange) throws IOException {
                    String requestedFilePath = exchange.getRequestURI().getPath();
                    System.out.println("Request incoming for: " + requestedFilePath);

                    File inFile = new File(requestedFilePath.substring(1));

                    if (inFile.exists()) {

                        String contentType = "text/html";
                                  if (requestedFilePath.endsWith(".css")) contentType =
"text/css";
                                  if (requestedFilePath.endsWith(".png")) contentType =
"image/png";

                        exchange.getResponseHeaders().add("Content-type", contentType);
                        exchange.sendResponseHeaders(200, inFile.length());

                        Files.copy(inFile.toPath(), exchange.getResponseBody());
                        exchange.getResponseBody().close();

                    } else {

                        String response = "<p>Hello Web Server...</p>";

                        exchange.getResponseHeaders().add("Content-type", "text/html");
                        exchange.sendResponseHeaders(200, response.length());

                        OutputStream responseStream = exchange.getResponseBody();
                        responseStream.write(response.getBytes());
                        responseStream.close();
                    }
                }
        } );
        server.start();
        System.out.println("Server has been started");
    }
```

```
}


========= EXPLANATION =========

-------------------------------
 IMPORTS
-------------------------------
These lines import Java tools your program needs:
- java.io & java.nio.file: Read/write files and send data to browser
- java.net: Set up your server on port 8000
- com.sun.net.httpserver.*: Built-in Java mini web server classes

-------------------------------
 CLASS AND MAIN METHOD
-------------------------------
public class WebServer {
This is the container for your whole program.

public static void main(String[] args) throws IOException {
This is where your program starts. throws IOException means the code might fail with
file/network errors.

-------------------------------
 CREATE SERVER
-------------------------------
HttpServer server = HttpServer.create(new InetSocketAddress(8000), 0);
Creates a web server on localhost:8000. 0 means no special backlog limit.

-------------------------------
 REGISTER ROUTE
-------------------------------
HttpContext context = server.createContext("/");
This means: for any path like /index.html or /style.css, well handle it below.

-------------------------------
 SET REQUEST HANDLER
-------------------------------
context.setHandler( new HttpHandler() {
You define what happens when a request comes in.

@Override
public void handle(HttpExchange exchange) throws IOException {
This function runs whenever a browser sends a request. 'exchange' is like a message
container.

-------------------------------
 READ REQUEST PATH
-------------------------------
String requestedFilePath = exchange.getRequestURI().getPath();
System.out.println("Request incoming for: " + requestedFilePath);
This reads the URL requested by the browser and prints it for debugging.

-------------------------------
```

```
 OPEN FILE IF EXISTS
-------------------------------
File inFile = new File(requestedFilePath.substring(1));
Tries to open the file requested by removing the leading "/".

if (inFile.exists()) {
If the file exists, well send it back.

-------------------------------
 SET MIME TYPE
-------------------------------
String contentType = "text/html";
if (requestedFilePath.endsWith(".css")) contentType = "text/css";
if (requestedFilePath.endsWith(".png")) contentType = "image/png";
Sets the type of the file for the browser to display correctly.

exchange.getResponseHeaders().add("Content-type", contentType);
exchange.sendResponseHeaders(200, inFile.length());
Tells browser: "OK, here comes a file of this size."

Files.copy(inFile.toPath(), exchange.getResponseBody());
exchange.getResponseBody().close();
Sends the file to the browser and closes the connection.

-------------------------------
 FILE NOT FOUND: DEFAULT MESSAGE
-------------------------------
} else {
String response = "<p>Hello Web Server...</p>";
This is a fallback HTML response.

exchange.getResponseHeaders().add("Content-type", "text/html");
exchange.sendResponseHeaders(200, response.length());
Tells the browser its an HTML response and how big.

OutputStream responseStream = exchange.getResponseBody();
responseStream.write(response.getBytes());
responseStream.close();
Writes the message and closes the stream.

-------------------------------
 START THE SERVER
-------------------------------
});
server.start();
System.out.println("Server has been started");
Starts the server and begins listening for requests.

-------------------------------
 SO WHAT IS A SERVER HERE?
-------------------------------
It's your Java program running a loop:
- Waits for connections
- Checks if the file exists
```

- Sends file or message back
- Repeats for every visitor to localhost:8000